Appendix D

PERFORMANCE
WIZARD

proliant

Introduction

Based on data collected in the host proliant, from 08/03/2001, at 00:00, to 08/10/2001, at 23:00, the current performance analysis report was elaborated.

The data used in this report was obtained from an exclusive collector, developed specially for this end, executing on the target machine with high resolution and low intrusion. This collector obtains data directly from the operating system, without any other libraries or additional tools, with a minimum overhead on the system. The data collected is stored using a binary format, in order to provide persistence. When automatically sent, it is compressed and encrypted, to ensure fast delivery and confidentiality.

The content of this report is based on years of experience in performance analysis and capacity planning. The tool used to generate this report operates in a completely automatic way, without direct human intervention. It uses an extensible inference machine, based on heuristics and rules, and is subject to continuous improvements. Using concepts such as "watermarks" and tolerance, it is possible to determine if a computational resource usage is excessive and if the excess is relevant.

During the monitoring period, the summary configuration of the machine, which has been obtained dynamically, was:

OS	:	MS Windows 2000 Advanced
Version	:	5.0.2195 (sp 2.0) Service Pack 2
Host	:	proliant
IP address	:	192.168.1.18
Processors	:	1 Pentium III (Coppermine)
Speed	:	728 MHz
Memory	:	191 MB

The last boot of the host proliant took place on 08/07/2001, at 10:32.

This report is based on monitoring which occurred between 08/03/2001, at 00:00, and 08/10/2001, at 23:00. The following was worth highlighting within this period:

For most of the time, CPU usage remained above 75%, reaching the maximum of 99.4%, and causing a bottleneck in the system.

A bottleneck was caused by the runnable process queue, because it exceeded the number of active processors for most of the time.

The average paging rate was high all the time, reaching 185.1 pps and causing a strong memory constraint.

During the monitoring period all the real memory was used for processes and file caching.

The high amount of virtual memory in use, during all the monitoring period, indicates a need for more real memory.

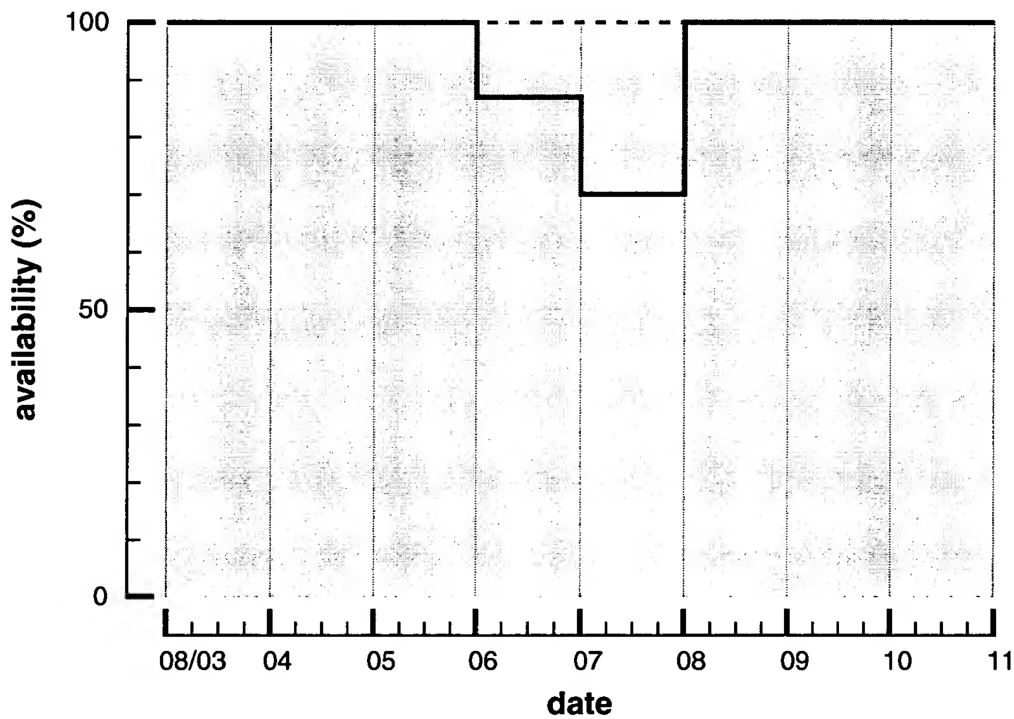
The network bandwith was always sufficient, not indicating any restraints.

For over 25% of the monitoring period, disk hdisk0 exceeded the usage limit.

The paging space usage exceeded the level of 70% for more than 50% of the time, reaching a maximum of 93.5%, thus resulting in a bottleneck in the system.

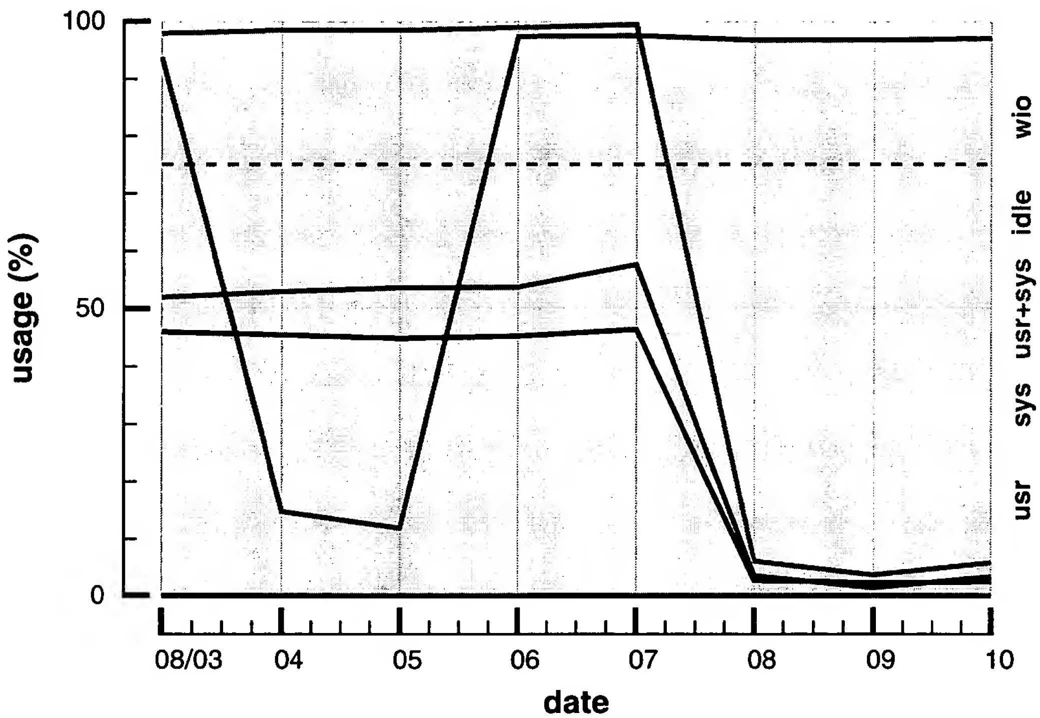
Availability

During this monitoring period the machine maintained the following availability rate.

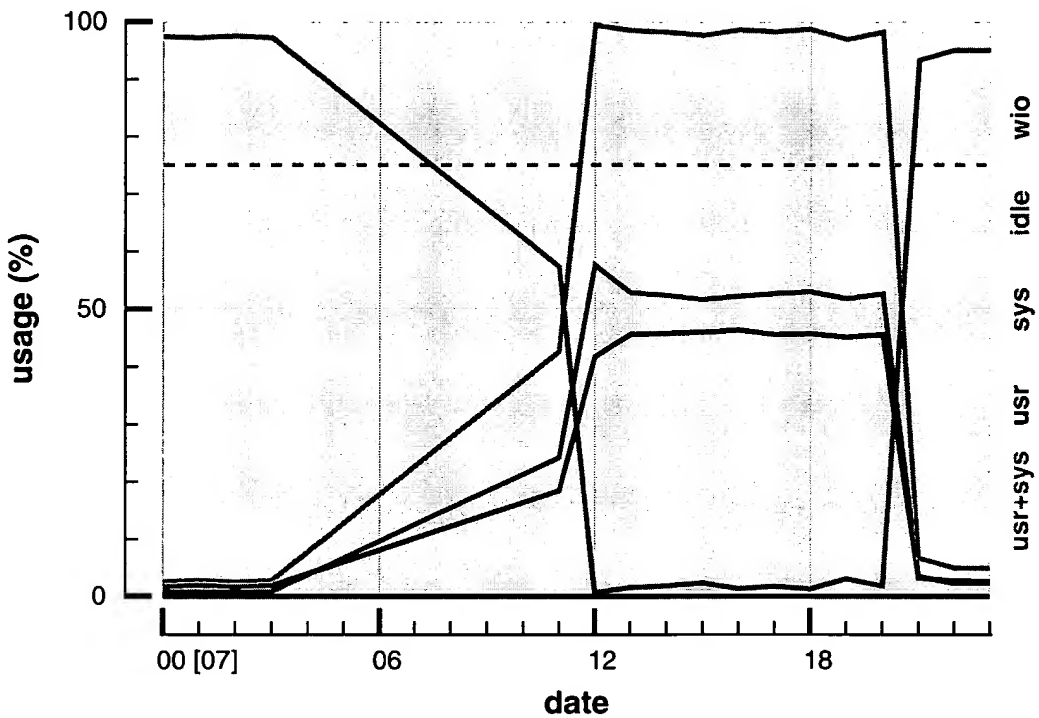


CPU Usage

CPU usage exceeded the level of 75% for most of the time, as shown in the graph below, thus resulting in a bottleneck in the system.



On 08/07, at 12:00,the CPU reached its maximum recorded usage. The graph below refers to this day.



These 10 processes were the ones which used the CPU the most on 08/07, at 12:00, when the highest level of usage was recorded.

ABSOLUTE

PROCESS	PID	THREADS	USAGE
sqlservr	944	36	9.8%
CSRSS	232	15	7.0%
CMD	1568	1	1.6%
CMD	2584	1	1.4%
CMD	780	1	1.4%
CMD	1780	1	1.3%
LSASS	292	20	1.3%
CMD	2744	1	1.3%
explorer	2044	11	0.9%
Idle	0	1	0.6%

GROUP

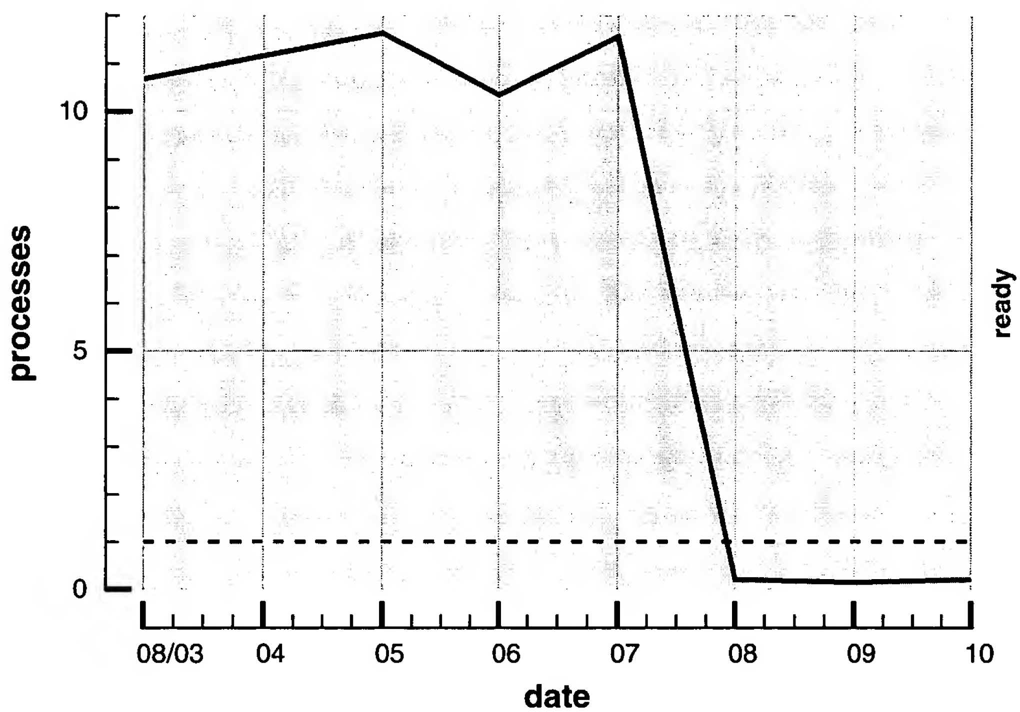
PROCESS	THREADS	USAGE
sqlservr	67	9.9%
CMD	7	7.1%
CSRSS	15	7.0%
LSASS	20	1.3%
explorer	11	0.9%
Idle	1	0.6%
cqmghost	9	0.4%
osql	28	0.3%
jre	137	0.3%
System	46	0.3%

Process Queue

For most of the monitoring period the runnable process queue exceeded the level of 1, which resulted in a bottleneck, as the graph below demonstrates.

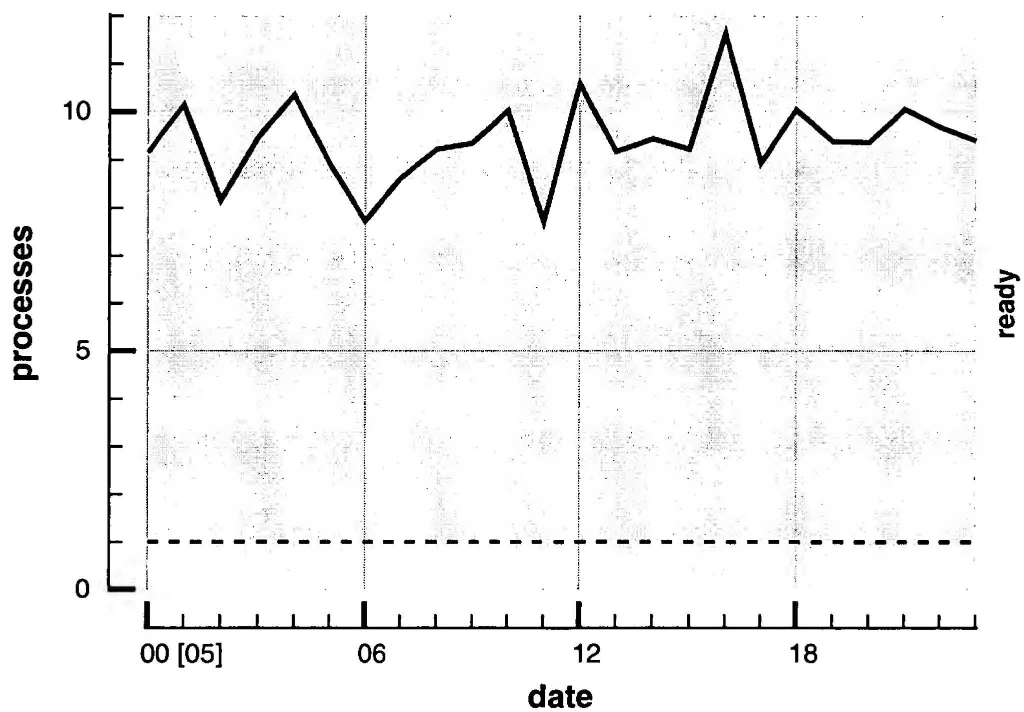
The greatest number of processes running was 64. This happened on 08/03, at 00:00.

The maximum number of running threads, 714, occurred on 08/04, at 23:00.



The runnable process queue reached its highest point on the 08/05, at 16:00.

At this exact moment, 54 processes and 681 threads were simultaneously running.



These 10 processes were the ones which most consumed the CPU on 08/05, at 16:00, when the runnable processes queue reached its highest number.

ABSOLUTE

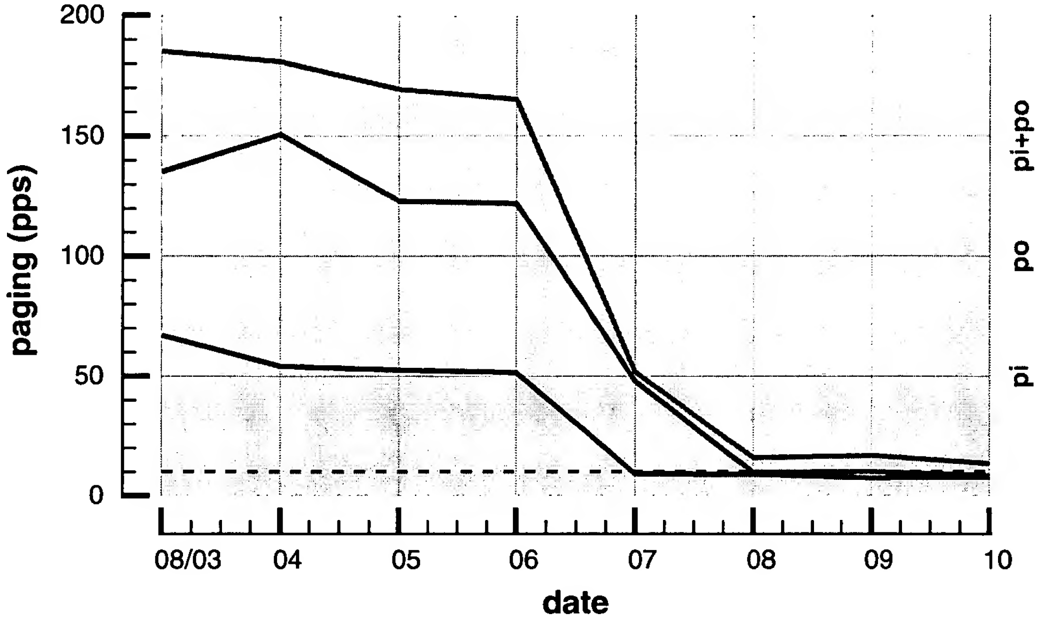
PROCESS	PID	THREADS	USAGE
sqlservr	2040	44	9.9%
CSRSS	232	20	7.9%
Idle	0	1	2.6%
jre	784	150	2.2%
System	8	47	1.9%
CMD	1720	1	1.6%
CMD	3232	1	1.4%
CMD	2812	1	1.4%
CMD	1140	1	1.4%
CMD	1476	1	1.4%

GROUP

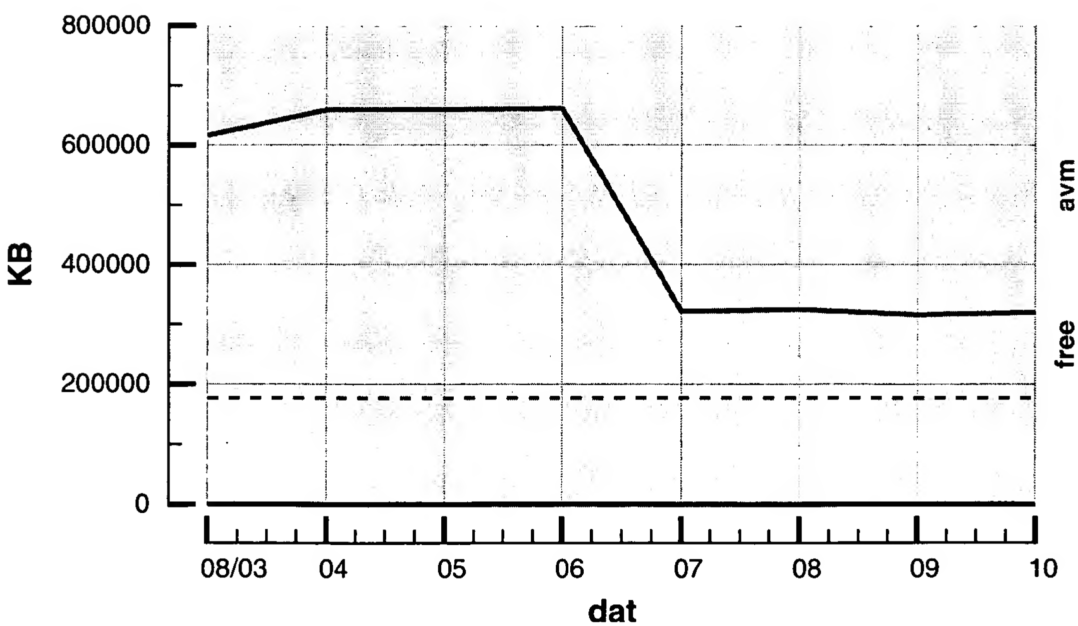
PROCESS	THREADS	USAGE
sqlservr	77	10.0%
CSRSS	20	7.9%
CMD	7	7.3%
Idle	1	2.6%
jre	150	2.2%
System	47	1.9%
LSASS	28	1.2%
cqmghost	9	0.3%
osql	16	0.3%
explorer	9	0.1%

Memory

The average paging rate was high all the time, indicating strong memory constraint.

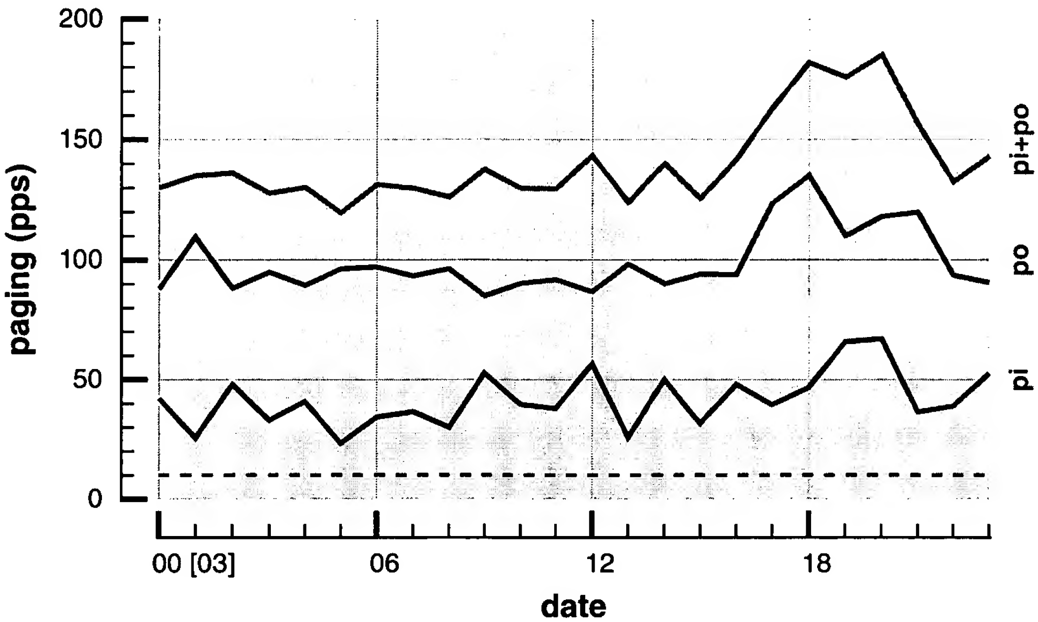


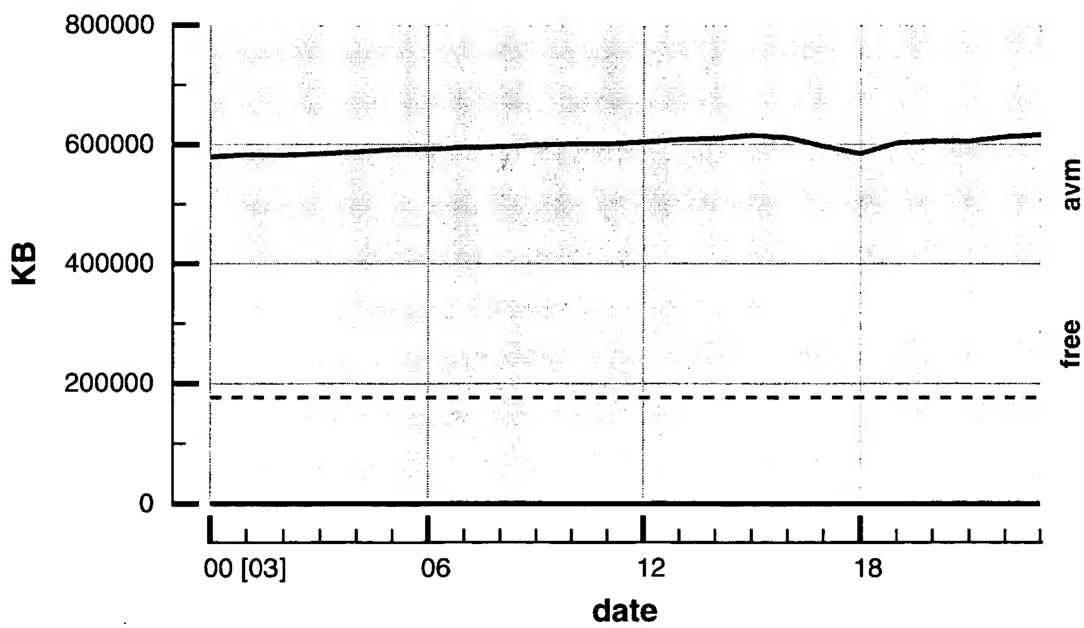
The virtual memory in use was high during all the recorded period.



All the real memory was used in processes and file caching during the monitoring period.

The graph below represents the date 08/03. On this day, at 20:00, memory reached the highest level of usage.



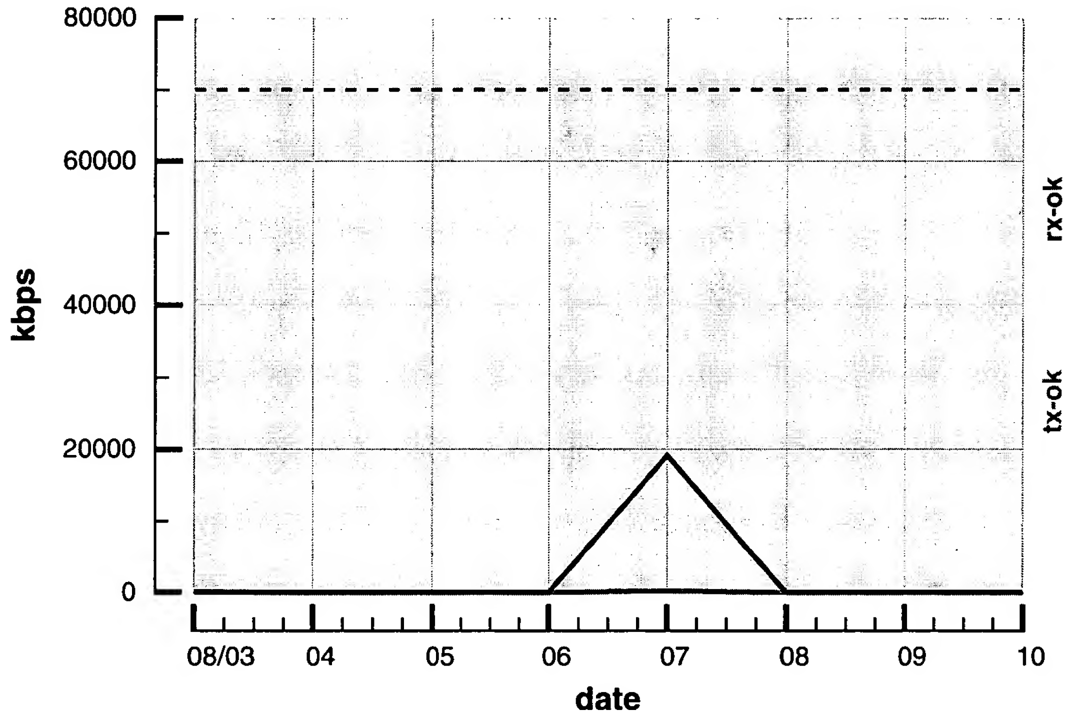


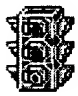
On 08/03, at 20:00, the highest level of paging occurred. These are the 10 processes that consumed most of the memory at this moment. Usage is shown in KB.

PROCESS	PID	THREADS	USAGE
jre	784	151	430,731
sqlservr	2040	37	43,836
sqlservr	828	32	13,240
inetinfo	1588	27	6,100
DLLHOST	2436	26	2,900
explorer	2656	11	2,929
surveyor	1044	7	2,680
SERVICES	280	34	2,897
Win32sl	1188	15	1,220
LSASS	292	18	2,626

There was no interface overload. Both transmission and reception rates remained below 70%.

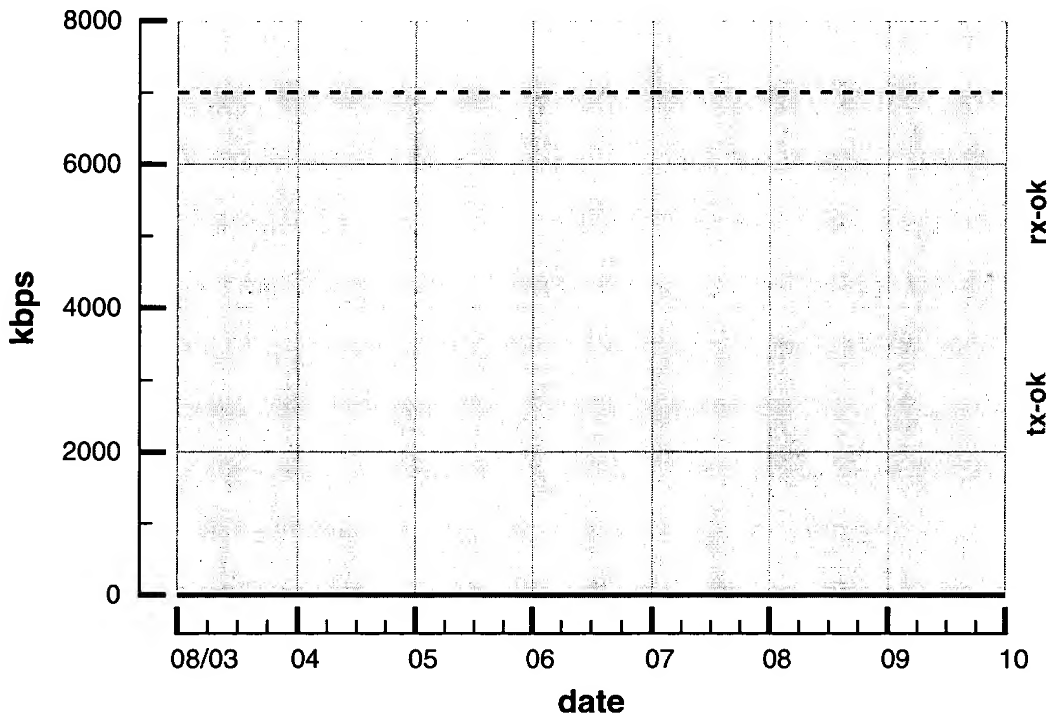
There were no errors in transmission and/or reception.



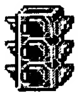


There was no interface overload. Both transmission and reception rates remained below 70%.

There were no errors in transmission and/or reception.



Disk I/O

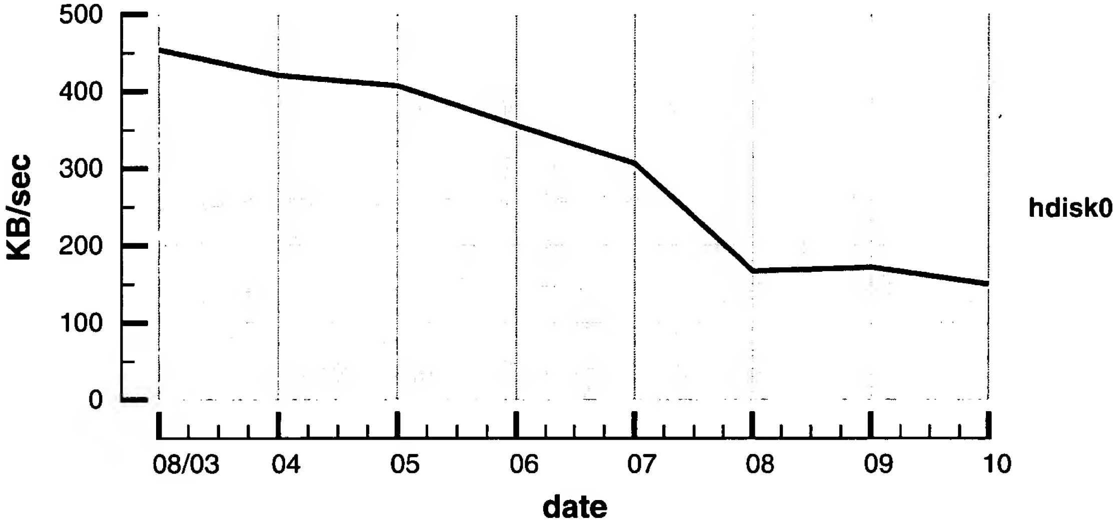


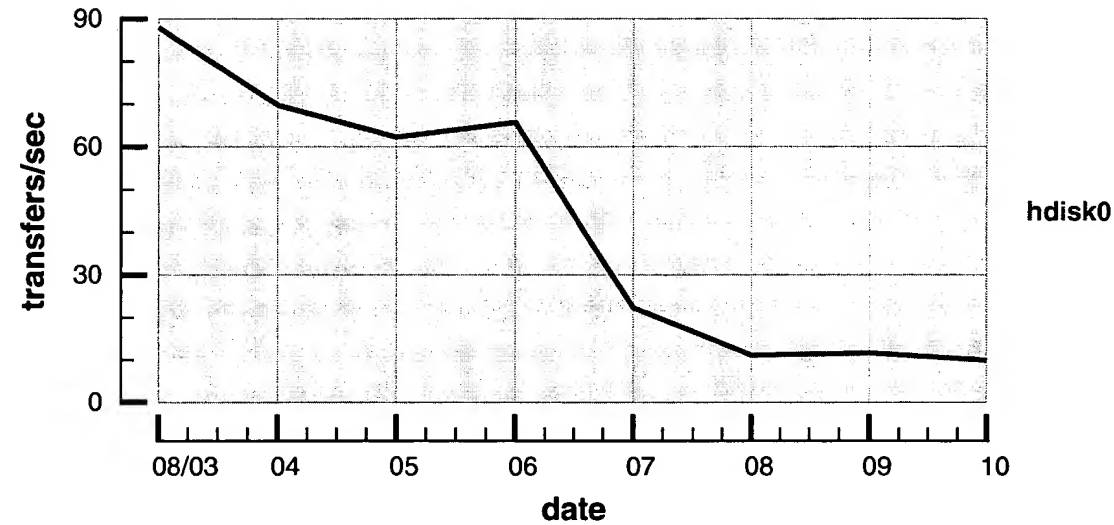
The disk hdisk0 was analyzed for the following requisites: transaction rate, transfers per second and usage.

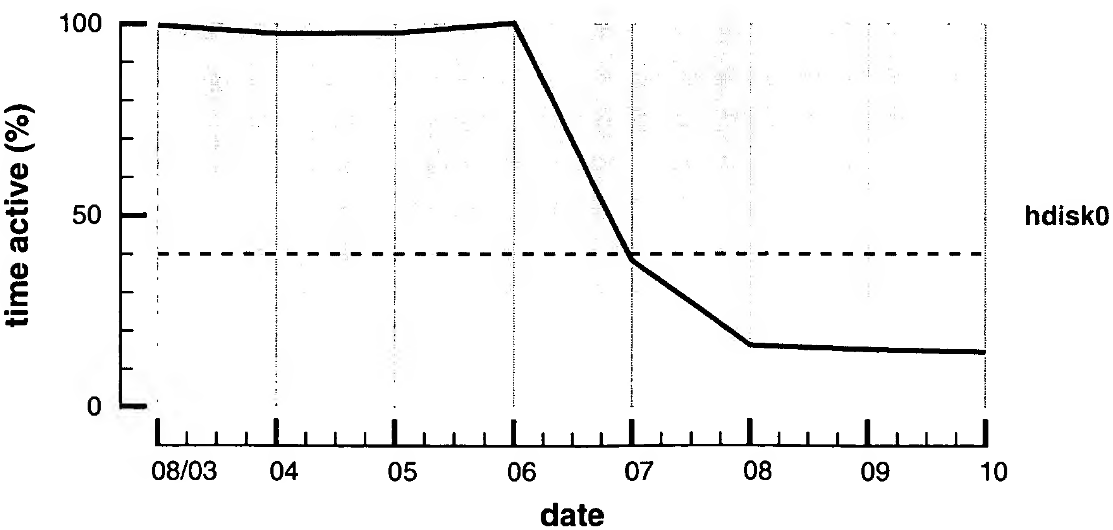
For over 25% of the monitoring period, disk hdisk0 exceeded the usage limit.

The graphs relating to disk hdisk0 are shown on the following pages.

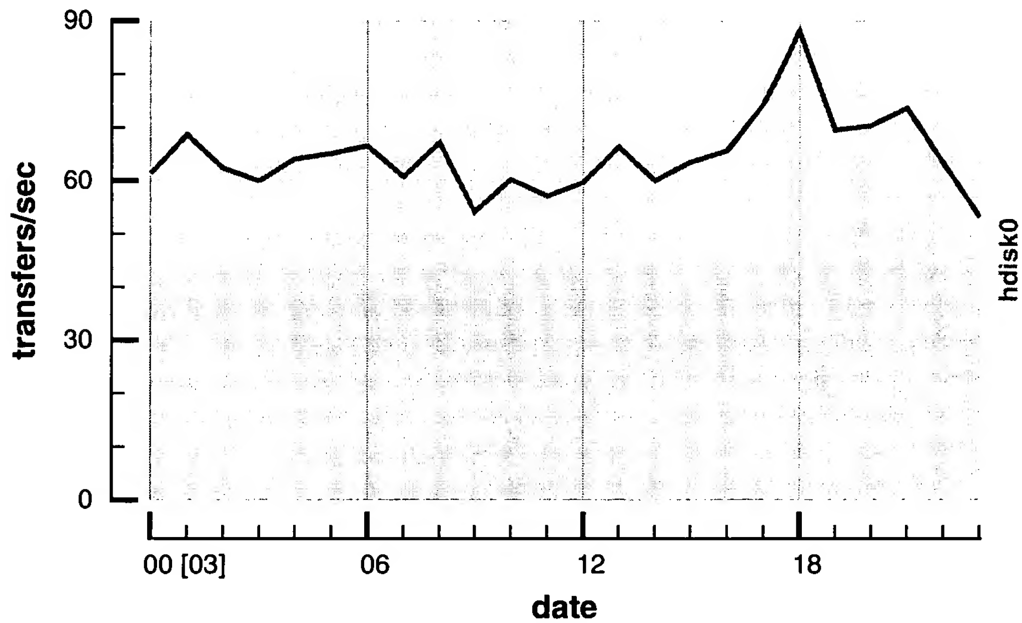
Disk I/O

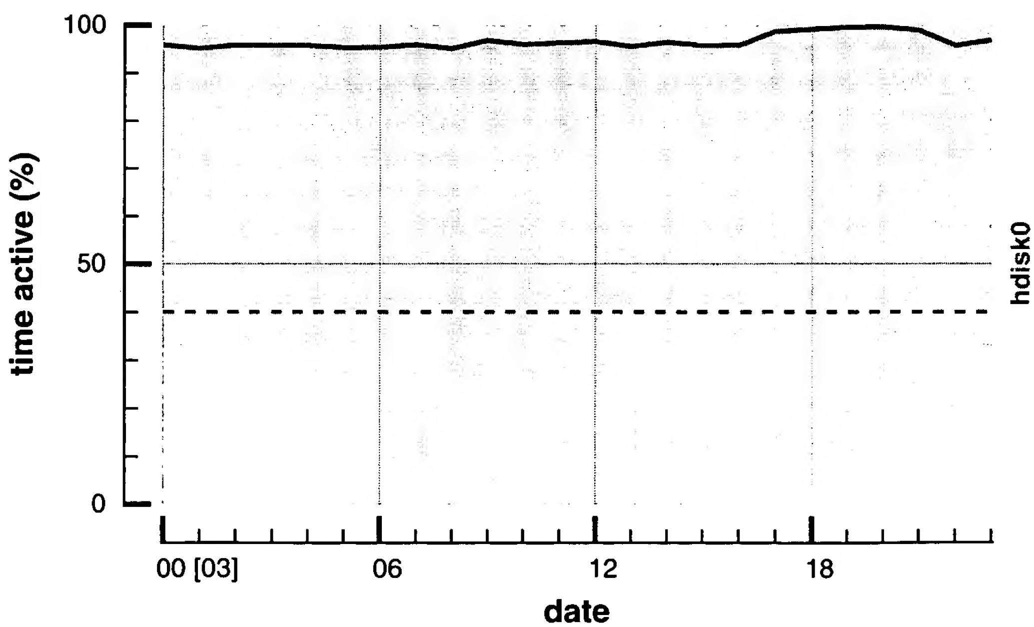

500
400
300
200
100
0
KB/sec
08/03
04
05
06
07
08
09
10
date
hdisk0


90
60
30
0
transfers/sec
08/03
04
05
06
07
08
09
10
date
hdisk0

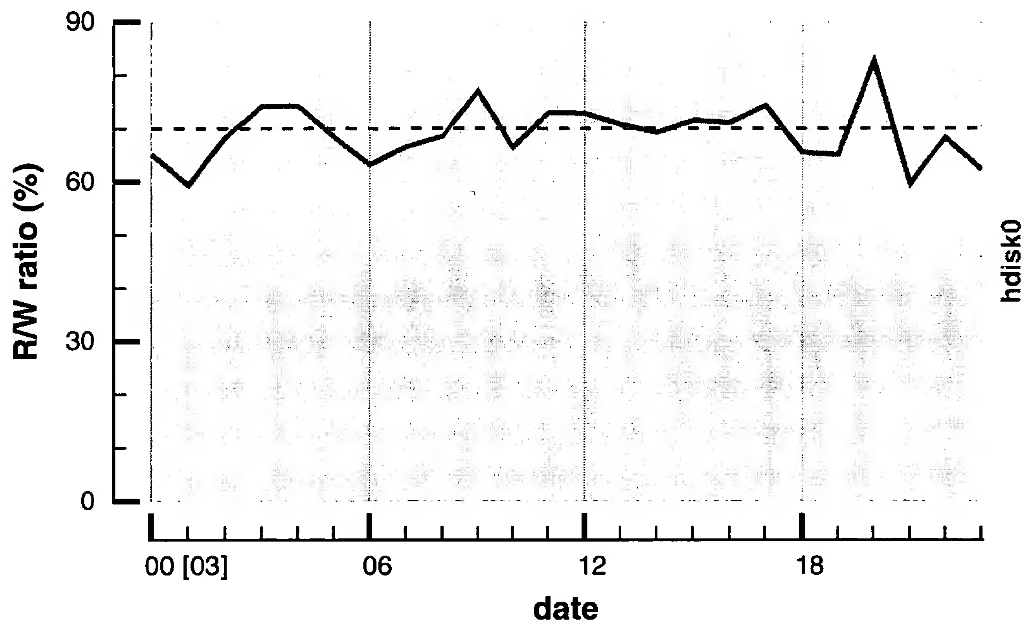

100
50
0
time active (%)
08/03
04
05
06
07
08
09
10
date
hdisk0

On 08/03, at 18:00, disk I/O showed the highest activity of the whole monitoring period. The graph below represents the status on this day.





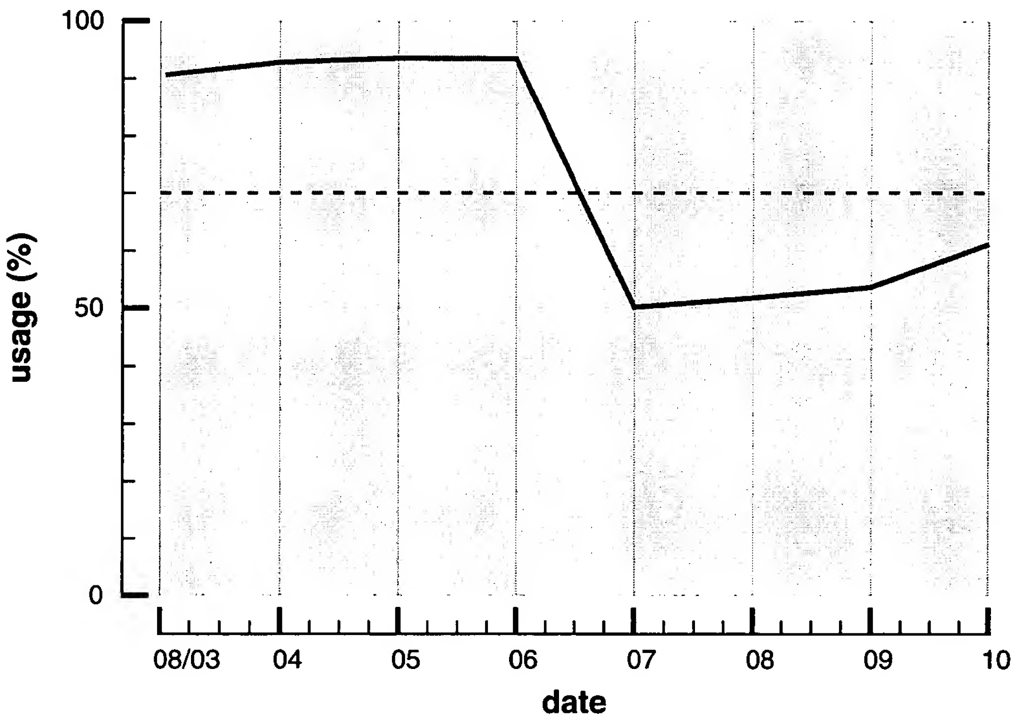
The graph below shows the disk hdisk0 Read/Write ratio, during the highest I/O activity period.



Paging Spac

The paging space usage, 294,912 KB, exceeded the level of 70%, as shown in the graph below, reaching a maximum of 93.5%, thus resulting in a bottleneck of the system.

The occupation rate trend for paging space was not analyzed due to the short sample period.



File System

Status of the file system at the end of the monitoring period:

MountPoint	PhysDrv	Type	Total (KB)	Free (KB)	%Used
D:\		NTFS	4,707,012	3,241,196	31
C:\		NTFS	4,136,737	886,382	78

Disk Occupation

This was the situation of the disk at the end of the monitored period:

Disk_Number	Signature	Size (MB)	Free (MB)	%Used
0	286383549	8,675	0	100

Processes that most used CPU during the monitoring period, as from 03/08.

ABSOLUTE

PROCESS	PID	THREADS	USAGE
Idle	0	1	22.9%
CSRSS	232	20	7.4%
sqlservr	856	43	6.8%
jre	784	165	2.3%
sqlservr	2040	34	1.1%
CMD	2856	1	1.0%
CMD	764	1	1.0%
System	8	46	1.0%
CMD	2116	1	1.0%
CMD	2880	1	1.0%

GROUP

PROCESS	THREADS	USAGE
Idle	1	22.9%
sqlservr	75	8.0%
CSRSS	20	7.4%
CMD	8	5.8%
jre	165	2.3%
System	46	1.5%
LSASS	21	1.0%
cqmghost	9	0.4%
osql	25	0.3%
explorer	11	< 0.1%

ABSOLUTE

PROCESS	PID	THREADS	USAGE
sqlservr	2040	43	9.7%
CSRSS	232	20	7.4%
Idle	0	1	6.5%
jre	784	176	2.3%
CMD	1720	1	1.6%
CMD	2812	1	1.4%
CMD	3232	1	1.4%
CMD	1140	1	1.4%
CMD	1476	1	1.4%
System	8	46	1.3%

GROUP

PROCESS	THREADS	USAGE
sqlservr	75	9.8%
CSRSS	20	7.4%
CMD	7	7.4%
Idle	1	6.5%
jre	176	2.3%
System	46	1.6%
LSASS	28	1.3%
cqmghost	9	0.4%
osql	41	0.4%
explorer	9	< 0.1%

ABSOLUTE

PROCESS	PID	THREADS	USAGE
sqlservr	2040	43	9.7%
CSRSS	232	20	7.5%
Idle	0	1	6.0%
jre	784	174	2.3%
System	8	46	2.0%
CMD	1720	1	1.6%
CMD	2812	1	1.4%
CMD	3232	1	1.4%
CMD	1476	1	1.4%
CMD	1140	1	1.4%

GROUP

PROCESS	THREADS	USAGE
sqlservr	76	9.8%
CSRSS	20	7.5%
CMD	7	7.4%
Idle	1	6.0%
jre	174	2.3%
System	46	2.0%
LSASS	28	1.2%
cqmghost	9	0.4%
osql	73	0.4%
explorer	9	< 0.1%

ABSOLUTE

PROCESS	PID	THREADS	USAGE
Idle	0	1	19.0%
sqlservr	2040	44	8.3%
CSRSS	232	20	6.4%
jre	784	171	1.9%
System	8	46	1.7%
CMD	1720	1	1.4%
CMD	2812	1	1.2%
CMD	3232	1	1.2%
CMD	1140	1	1.1%
CMD	1476	1	1.1%

GROUP

PROCESS	THREADS	USAGE
Idle	1	19.1%
sqlservr	77	8.7%
CSRSS	20	6.4%
CMD	7	6.3%
jre	171	2.0%
System	46	1.7%
LSASS	28	1.1%
cqmghost	9	0.4%
osql	367	0.3%
explorer	9	< 0.1%

ABSOLUTE

PROCESS	PID	THREADS	USAGE
Idle	0	1	43.2%
sqlservr	944	35	6.5%
CSRSS	232	11	4.4%
CMD	1568	1	1.0%
CMD	2584	1	0.8%
LSASS	292	20	0.8%
CMD	780	1	0.8%
CMD	2744	1	0.8%
CMD	1780	1	0.8%
sqlservr	856	35	0.4%

GROUP

PROCESS	THREADS	USAGE
Idle	1	43.2%
sqlservr	66	7.1%
CMD	7	4.4%
CSRSS	11	4.4%
LSASS	20	0.8%
cqmghost	9	0.4%
explorer	11	0.3%
jre	135	0.3%
osql	28	0.2%
System	46	0.1%

ABSOLUTE

PROCESS	PID	THREADS	USAGE
Idle	0	1	95.6%
sqlservr	944	39	1.7%
explorer	2044	9	0.5%
sqlservr	828	33	0.5%
cqmghost	1740	9	0.3%
taskmgr	2056	3	0.3%
jre	796	136	0.3%
WinVNC	1480	5	0.2%
LSASS	292	18	0.2%
DLLHOST	2512	25	< 0.1%

GROUP

PROCESS	THREADS	USAGE
Idle	1	95.6%
sqlservr	72	2.2%
explorer	9	0.5%
cqmghost	9	0.3%
taskmgr	3	0.3%
jre	136	0.3%
WinVNC	5	0.2%
LSASS	18	0.2%
DLLHOST	35	< 0.1%
inetinfo	29	< 0.1%

ABSOLUTE

PROCESS	PID	THREADS	USAGE
Idle	0	1	96.6%
sqlservr	944	39	1.7%
sqlservr	828	33	0.5%
cqmghost	1740	9	0.3%
jre	796	135	0.3%
LSASS	292	18	0.2%
WinVNC	1480	5	< 0.1%
DLLHOST	2512	25	< 0.1%
inetinfo	1632	28	< 0.1%
explorer	2044	9	< 0.1%

GROUP

PROCESS	THREADS	USAGE
Idle	1	96.6%
sqlservr	72	2.2%
cqmghost	9	0.3%
jre	135	0.3%
LSASS	18	0.2%
WinVNC	5	< 0.1%
DLLHOST	35	< 0.1%
inetinfo	28	< 0.1%
explorer	9	< 0.1%
aengine	6	< 0.1%

ABSOLUTE

PROCESS	PID	THREADS	USAGE
Idle	0	1	96.5%
sqlservr	944	39	1.7%
sqlservr	828	33	0.5%
cqmghost	1740	9	0.3%
jre	796	135	0.3%
WinVNC	1480	5	0.2%
LSASS	292	18	0.1%
DLLHOST	2512	25	< 0.1%
inetinfo	1632	29	< 0.1%
System	8	46	< 0.1%

GROUP

PROCESS	THREADS	USAGE
Idle	1	96.5%
sqlservr	72	2.2%
cqmghost	9	0.3%
jre	135	0.3%
WinVNC	5	0.2%
LSASS	18	0.1%
DLLHOST	35	< 0.1%
inetinfo	29	< 0.1%
aengine	6	< 0.1%
System	46	< 0.1%

Processes that most used memory during the monitoring period, as from 08/03. The usage is shown in KB.

ABSOLUTE

PROCESS	PID	THREADS	USAGE
jre	784	165	433,005
sqlservr	856	43	48,601
sqlservr	2040	34	45,324
sqlservr	828	32	13,240
inetinfo	1588	27	6,100
WINLOGON	252	15	5,480
mmc	2896	3	4,750
aengine	652	6	3,894
aengine	836	6	3,774
explorer	2656	11	2,926

ABSOLUTE

PROCESS	PID	THREADS	USAGE
jre	784	176	472,577
sqlservr	2040	43	47,673
sqlservr	828	32	13,247
WINLOGON	252	15	5,480
aengine	836	6	3,758
LSASS	292	28	2,970
SERVICES	280	34	2,909
cpqwmgmt	1864	5	2,568
snmp	960	10	2,412
Webdmi	1132	8	2,332

ABSOLUTE

PROCESS	PID	THREADS	USAGE
jre	784	174	473,016
sqlservr	2040	43	46,765
sqlservr	828	33	13,323
WINLOGON	252	15	5,480
aengine	836	6	3,765
LSASS	292	28	2,966
SERVICES	280	35	2,908
surveyor	1044	7	2,680
cpqwmgmt	1864	5	2,568
arelay	648	6	2,444

ABSOLUTE

PROCESS	PID	THREADS	USAGE
jre	784	171	471,122
sqlservr	856	32	107,719
sqlservr	2040	44	47,548
jre	792	136	42,706
IEXPLORE	2696	17	17,022
sqlservr	824	31	13,327
sqlservr	828	33	13,325
inetinfo	1584	27	5,892
WINLOGON	252	15	5,494
aengine	836	6	3,820

ABSOLUTE

PROCESS	PID	THREADS	USAGE
sqlservr	856	35	124,860
sqlservr	944	35	116,700
jre	796	142	44,393
jre	792	135	43,765
sqlservr	824	31	13,360
sqlservr	828	31	13,339
IEXPLORE	2804	13	8,671
inetinfo	1584	27	6,072
inetinfo	1632	27	5,961
WINLOGON	252	15	5,480

ABSOLUTE

PROCESS	PID	THREADS	USAGE
sqlservr	944	39	125,897
jre	796	136	43,987
sqlservr	828	33	13,330
inetinfo	1632	29	6,131
WINLOGON	252	15	5,480
explorer	2044	9	4,594
mmc	2972	7	3,932
SERVICES	280	36	3,031
surveyor	1088	7	2,853
DLLHOST	2512	25	2,788

ABSOLUTE

PROCESS	PID	THREADS	USAGE
sqlservr	944	39	120,017
jre	796	135	43,987
sqlservr	828	33	13,333
inetinfo	1632	28	6,124
WINLOGON	252	15	5,480
explorer	2044	9	4,848
mmc	2972	7	3,932
SERVICES	280	37	3,094
surveyor	1088	7	2,875
DLLHOST	2512	25	2,843

ABSOLUTE

PROCESS	PID	THREADS	USAGE
sqlservr	944	39	119,802
jre	796	135	44,126
sqlservr	828	33	13,217
inetinfo	1632	29	6,109
WINLOGON	252	15	5,493
explorer	2044	9	4,839
mmc	2972	7	3,932
aengine	2048	6	3,804
SERVICES	280	37	3,078
DLLHOST	2512	25	2,884

In order to be able to understand a performance analysis report, it may be convenient to review some basic concepts. The idea here is not to make a treaty on the subject, but to go through some fundamental aspects related to performance.

System performance means different things to different people. This can range from resource consumption to amount of work performed per unit of time. It will be assumed here that improving performance means improving response time of end users and/or increasing throughput of both end user work and batch work.

The performance of any system depends on how tied up key resources are. The reason being that system performance is, essentially, a function of the time each key resource takes to service a request, plus the time a request has spent queued waiting to be serviced (more details on queues ahead). In case of an information-processing environment, based on computers, key resources are CPU, memory, disk I/O and network I/O.

In order to evaluate resource consumption, criteria must be established. These criteria consist of judging which system performance variables best express this consumption, since many are available. In addition, the watermarks (point where a resource starts to be considered overcommited - also known as thresholds) for these variables need be defined. These watermarks are approximate and can vary depending on the characteristics of the system being analyzed.

Description of key resources

1 - CPU

CPUs can play a significant role in the response time of computational environments, especially when other resources are abundant. This is particularly true in environments where most of the data required is available in memory. For CPU, the key variables to evaluate resource consumption are run-queue and CPU usage.

1.1 - Run-queue

Run-queue means the amount of processes (threads, in fact) which are runnable (ready to execute), being either queued, waiting for a CPU, or already executing. It is a measure of how used up is the CPU, in an environment comprised of many processes (a commercial transaction environment, e.g.). The watermarks, typical of run-queue, are a range between the number of processors available and five times this value. This depends on the response time required for a transaction, versus the amount of CPU required by this transaction.

1.2 - CPU Usage

CPU usage is a measure of how used up is the CPU in an environment

comprised of few heavy processes (such is the case of scientific or commercial environments with few, but complex, batches). It can be used as a criterion for environments with many processes, but run-queue is more meaningful in these cases. CPU usage is expressed in percentage and it can be broken in four categories, usr, sys, idle and wio. Usr stands for user mode or the mode in which a process executes, when not using any operating system service. Sys means system mode, which is the mode a process is placed into when using any operating system service. Idle, as the term suggests, is when a CPU has no process to execute. Wio stands for waiting for I/O, a special case of idleness, where the CPU is available, but there are processes waiting for an I/O operation to complete. CPU usage is normally a concern when usr+sys is above 75 to 85%, in an environment with multiple processes, or is close to 100% / number of processors, in an environment with few processes.

2 - Memory

Memory can play different roles in a computational environment, ranging from fast storage area for program data to disk data caching (making up for the slower speed of disk subsystems). This means that memory is consumed for very distinct purposes. Memory consumption, being understood as not only real memory (RAM), but as the entire virtual memory subsystem, can be well evaluated by paging activity, virtual memory usage and paging space usage.

Paging activity occurs when the real memory being managed by a virtual memory subsystem is overcommitted. In a small degree, it is not a problem, since the main purpose of the virtual memory subsystem is to be able to maximize system throughput by allowing process memory to be swapped in and out. It becomes a critical issue when paging reaches high rates. The point is that paging indicates that the sum of the working sets (ranges of virtual memory addresses of processes that need to be accessible at a given moment) of the processes, plus what is left aside for the operating system and file caching, exceeds the amount of real memory available. Paging is broken down in page in (pi) and page out (po). A page in is usually regarded as more serious, since it may indicate a thrashing condition (the system is spending too much time just paging). The watermark for paging (pi+po) is in the range of 10 pages per second.

Paging space usage is a fundamental concern when analyzing the state of the virtual memory manager. If no paging space is available, definitely no new process will be spawned and, very likely, some existing processes may be terminated by the operating system in order to make room in the paging space. So, real memory constraints impact performance, but paging space constraints put in risk the entire execution environment. The amount of virtual memory devoted to process segments (process data area) is directly related to paging space usage, if the operating system in question is working with early paging

space allocation (allocate space in the paging area whenever it is allocated in real memory). In this case, the amount of space being used in the paging area is the sum of the data areas of all running processes, which is the major component in determining the amount of real memory required by the system. If the amount of virtual memory in use exceeds the amount of real memory available, paging will very likely occur, initiating peformance degradation. In an environment experiencing significant growth rates, especially in terms of users, it is advisable to keep the average paging space usage rate at 50%. Obviously, this concern does not exist or is less serious in the case of operating systems that are able to allocate paging space dynamically.

3 - Disk I/O

Disk I/O is certainly one of the main subjects when performance is in discussion. This is particularly true in commercial environments. Disks, being mechanical devices (in comparison with other devices which are faster, because they are eletronic) can, if not properly used, put in jeopardy the performance of an entire system. In addition, disks can present two very distinct personalities - one, when accessing data in random mode (slower, because it involves arm movement), and other, when accessing data in sequential mode (faster, because it only involves plate movement). Also, the performance of disks varies according to the blocking factor (amount of data involved in a same operation), since the impact of the overhead is dilluted, in the case of large blocks. Therefore, disks must be closely watched. Among the key variables that provide information on disk I/O usage are bandwidth occupation, transfers (I/Os) per second, transfer rate (usually expressed in KB/s) and physical read to write ratio.

3.1 - Bandwidth occupation

Bandwidth occupation is probably the most important variable when evaluating disk I/O. It is calculated based on the number of samples taken within a period (1 second, e.g.) that found a given disk to be busy. It is highly dependent on the rate of requests being sent to the disks and the type of data access being required by these requests (since random requests take longer to be serviced than sequential requests). With bandwidth occupation, it is possible to estimate whether disk I/O requests for a given disk are spending time in queues, instead of being serviced promptly. The disk bandwidth occupation watermarks regarded as acceptable vary from 15%, for environments with predominantly random access (OLTP with simple transactions), to 65%, for environments with predominantly sequential access (datawarehouse, complex batch applications, etc.). A criterion of 40% is adequate for mixed environments (which constitute the great majority of cases).

3.2 - Transfers per second

Transfers per second provide a good complementary information to

bandwidth occupation, specially in order to evaluate more subtle bottlenecks such as disk adapters (SCSI, FC-AL, SSA, etc.). Disks adapters have a ceiling, in terms of transfers per second, that might be reached without notice, limiting, therefore, the I/O capability of multiple disks. Physical disks support about 100 to 120 transfers per second, when accessing data in random mode, and 10x these values or more, when accessing data in sequential mode. So, it is considered acceptable to keep disks operating at a sustained rate of about 50% of these values.

3.3 - Transfer rate

Transfer rate informs the amount of data that is being received from disks or being sent to disks, per unit of time. Like bandwidth occupation and transfers per second, transfer rate is a function of the rate of requests being sent to the disks and the type of access being required by these requests (random access requires more time to be serviced and, therefore, limits transfer rates). Another key aspect of transfer rate is that it may also expose the ceiling of disk adapters, regarding this characteristic. In addition, computer I/O buses may also impose a limit on the transfer rates of adapters inserted in these buses. The typical watermarks for transfer rates, per individual physical disk, varies from 400 to 1.000 KB/s, for random I/O, to between 4.000 KB/s and 25.000 KB/s (when using large blocks), for sequential I/O.

3.4 - Physical read to write ratio

The physical read to write ratio for disk I/O is important in determining whether a given database configuration is adequate and whether the type of disk layout being used is the most suitable. When the number of physical reads more than exceeds 5x the number of writes, it might mean (although not necessarily) that the size of a database buffer cache is not big enough. Therefore, the database software might be operating with an unacceptable hit ratio (ratio of logical reads that are satisfied by the cache). A physical read to write ratio below 2.5x, although not a problem in itself, might not be adequate for certain disk layouts such as RAID-5, since this arrangement has a significant write-penalty (additional effort required to perform write operations, when compared to read operations).

4 - Network I/O

Network I/O, although also a key resource, seldomly plays a major role in influencing response time, except when wide area networks (WAN's) are involved. Nevertheless, this resource must also be monitored, for it may hide some surprises. The key variables to evaluate network I/O are transfer rate (or, bandwidth occupation of the maximum transfer rate), error rate and latency.

4.1 - Transfer rate

Similar to disk I/O, the transfer rate of network adapters depends on block size, although the impact is not as significant. On the other hand, there is direct correlation between transfer rate and bandwidth usage in the case of network I/O, whereas with disk I/O this is not the case (since the data access mode must be considered). Typically, collision-prone networks (Ethernet, without switches, e.g.) should operate at 30 to 40% of their nominal capacity and other types of networks should be kept at 50 to 70% of their nominal capacity.

4.2 - Error rate

Error rate provides a measure of how effective the transfer rate is in a network, since a high error rate will mean that the effective transfer rate is very low. Most network adapters and network device drivers provide some means of retrieving error information, which is presented as a complementary statistic to the transfer rate itself. Major causes of high error rates in LANs are collisions (two or more network devices trying to send data at the same time), stationary waves (signal that remains in the network due to poor termination), full-duplex/half-duplex mismatch (operation mode mismatch between adapters and hubs/swiches) and speed mismatch (speed mismatch between adapters and hubs/switches). A major cause of high error rates in WANs is noise (heat, electromagnetic noise, etc.). The watermark for error rate in LANs should be of 1% or even less and for WANs should be of about 5%.

4.3 - Network latency

Network latency can be measured by various schemes, the most common being echo requests (TCP/IP ping, e.g.). It is a measure of how long the first packet, of a chain of packets, took to reach its destination. The point is that it is not enough to have high bandwidth if the first packet requires a very significant time to arrive. This is the case with satellite links, but may also apply to confined networks. For instance, the latency of ATM and gigabit Ethernet can be very small for conventional applications, but it is high for parallel computing applications.. Another aspect of latency is that it may have an important software component, such as the one caused by operating systems protocol stacks (communication subsystems). The typical latencies desired for corporate LAN's (Local Area Networks) are in the range of 1 to 10 ms.

Automatos does not guarantee that the recommendations in this report are necessarily the best opportunities available in the market for the specific needs of the client. Automatos will not be held responsible for any kind of damages or losses relating to the recommendations, either directly, indirectly, punitively, incidentally, specially or consequently, including, but not limited to, loss of functions, data or profit, regardless of any contractual, non-contractual or objective obligation. Automatos will not be held responsible even if previously informed of the possibility of damage. The decision on which investment, product or

service to use is a sole responsibility of the client.